IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant:	Robin Alexis Takasugi et al.	Examiner: Sheng Jen Tsai
Serial No.:	10/672,975	Group Art Unit: 2186
Filed:	September 26, 2003	Docket No.: 10014268-1 / H303.154.101
Due Date:	**June 14, 2008**	
Title:	PREFETCH CONTROLLER FOR CONTROLLING RETRIEVAL OF DATA FROM A DATA STORAGE DEVICE	

REPLY BRIEF TO EXAMINER'S ANSWER

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir/Madam:

This Reply Brief is presented in response to the Examiner's Answer mailed April 14, 2008, and in support of the Notice of Appeal filed on November 9, 2006, and the Appeal Brief filed January 9, 2007, appealing the rejection of claims 1-30 of the above-identified application as set forth in the Final Office Action mailed August 9, 2006.

At any time during the pendency of this application, please charge any fees required or credit any overpayment due to Deposit Account No. 08-2025 pursuant to 37 C.F.R. 1.25. Additionally, please charge any fees required to Deposit Account No. 08-2025 under 37 C.F.R. 1.16, 1.17, 1.19, 1.20 and 1.21.

Appellant respectfully requests reconsideration and reversal of the Examiner's rejection of pending claims 1-30.

ARGUMENT

The language and arguments set forth in the Grounds of Rejection section (Examiner's Answer at pages 3-12) appear to be substantially similar to the language and arguments set forth in the Final Office Action mailed August 9, 2006. Appellant has addressed the deficiencies in these rejections in the Appeal Brief submitted on January 9, 2007. Therefore, the following remarks will be directed to the Examiner's arguments in the Response to Arguments section, which begins on page 14 of the Examiner's Answer.

Appellant pointed out in the Appeal Brief filed on January 9, 2007 that Hicken does not teach or suggest adding a prefetch value to a transfer length value specified in a current non-sequential read command, and then providing this sum to a data storage device, thereby requesting data specified by the current host command and prefetch data. Appellant addressed the Examiner's citations in detail in the Appeal Brief, and showed that Hicken fails to include such a teaching or suggestion. Now for the first time, the Examiner has argued in the Response to Arguments section that "[i]t is inherent that 'an adding function' has to be performed in order to 'adding a prefetch value to a transfer length value specified in a current non-sequential read command, and then providing this sum to a data storage device,' as recited in claim 1." (See, e.g., Examiner's Answer at page 19). It is not clear to Appellant, but the Examiner appears to be arguing that the limitations have to be inherent in Hicken because the limitations are present in the claim.

Not only is the Examiner's new inherency argument an implicit admission that there is no explicit teaching or suggestion in Hicken regarding the above limitations, it is clearly improper and contrary to established precedent. As the Federal Circuit has stated, "[i]nherent anticipation requires that the missing descriptive material is 'necessarily present,' not merely probably or possibly present, in the prior art." *Trintec Indus., v. Top-U.S.A. Corp.*, 63 USPQ2d 1597, 1599 (Fed. Cir. 2002) (quoting *In re Robertson*, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999)). It is clear that Hicken can perform all of the functions recited in that reference without also having to perform the adding function recited in claim 1. The Examiner has not provided any rational underpinning to support the argument that Hicken must necessarily add a prefetch value to a transfer length value specified in a current non-sequential read command. In fact, the numerous disclosures throughout Hicken that indicate that the prefetch is a separate transfer, teach away from performing such an adding function.

Thus, Hicken does not explicitly teach or suggest each and every limitation of the independent claims, and the missing claim limitations are not "necessarily present" in Hicken. Each of the Examiner's specific responses is addressed in further detail below.

The Examiner stated in the Response to Arguments section with respect to claims 1-8 that:

> Appellants contend that claims 1-8 are patentable over Hicken et al. (US 6,092,149, hereinafter referred to as Hicken) as, allegedly, Hicken fails to teach, or suggest "adding a prefetch value to a transfer length value specified in a current non-sequential read command, and then providing this sum to a data storage device." The Examiner disagrees with this argument for the following reasons:
>
> **First**, figure 1F of Hicken shows that the transferred data comprises the "requested data" and the "prefetched data," and the transfer length value is the sum of the length of the requested data and the length of the prefetched data. This clearly suggests that the length of data to be transferred is the sum of the length of the "requested data" and the length of the "prefetched data." (Examiner's Answer at page 14) (emphasis in original).

There is no disclosure in Figure 1F or its corresponding description, nor anywhere else in Hicken, that teaches or suggests adding a prefetch value to a transfer length value specified in a current read command, and then providing this sum to a data storage device.

The Examiner further stated in the Response to Arguments section with respect to claims 1-8 that:

> **Second**, figure 1D of Hicken also shows the segment associated with each cache entry comprises a LBA (Logic Base Address, also shown in figure 1F, 180) and a PF LBA (PreFetch Logic Base Address, also shown in figure 1F, 182). Since the caching system divides the cache memory into segments to store multiple streams of data and each segment is transferred as a unit [abstract]. This provides another piece of evidence that that the length of data to be transferred, as a segment, is the sum of the length of the "requested data" and the length of the "prefetched data." (Examiner's Answer at page 15) (emphasis in original).

There is no disclosure in Figures 1D or 1F or their corresponding descriptions, nor anywhere else in Hicken, that teaches or suggests adding a prefetch value to a transfer length value specified in a current read command, and then providing this sum to a data storage device. These Figures simply show how data is stored in the cache **after** it has been retrieved. The Examiner stated above that the "the length of data to be transferred, as a

segment, is the sum of the length of the 'requested data' and the length of the 'prefetched data'." The Examiner has not identified any disclosure in Hicken that teaches or suggests that the length of the requested data and the length of the prefetch data are actually summed or added. The Examiner has not identified any structure in Hicken that performs such an addition. The Examiner has not identified any disclosure in Hicken that teaches or suggests that such a sum is provided to a data storage device.

The Examiner further stated in the Response to Arguments section with respect to claims 1-8 that:

> **Third**, figure 1C of Hicken further shows the flow chart of the overall process. Significantly, it illustrates step 126 performs "compute prefetch" to determine the length of prefetched data, followed by step 132, which "set buffer counter and start the disk" to set buffer counter to be the total length of data to be transferred and then start the disk to transfer data to the disk. This provides yet another piece of evidence that that the length of data to be transferred, as a segment, is the sum of the length of the "requested data" and the length of the "prefetched data." (Examiner's Answer at page 15) (emphasis in original).

Figure 1C does not disclose "set buffer counter to be the total length of data to be transferred," as indicated by the Examiner. Regardless, there is no disclosure in Figure 1C or its corresponding description, that teaches or suggests adding a prefetch value to a transfer length value specified in a current read command, and then providing this sum to a data storage device.

The Examiner further stated in the Response to Arguments section with respect to claims 1-8 that:

> Appellants further argue that the prefetch in Hicken is a separate transfer that is determined after the requested data is retrieved, thus Hicken fails to teach, or suggest "adding a prefetch value to a transfer length value specified in a current non-sequential read command, and then providing this sum to a data storage device." The Examiner disagrees with this argument for the following reasons:
>
> First, Hicken's invention is directed toward dynamically prefetching data to maximize disk drive performance based on past access history [abstract]. Particularly, the amount of data to be prefetched for the current read command depends on the data acquired during the previous read commands as well as the type of access [figure 1G illustrates several types of access, including "full cache hit," "partial cache hit," "skip ahead cache hit," "sequential cache hit" and "no cache hit;" column 9, lines 28-67].

> For instance, if the data acquired in previous read commands (the prefetched data is stored in the cache buffer [figure 1A, 10]) already includes portion of the data that is requested by the current read command, then the amount of data to be prefetched for the current read command needs to be adjusted accordingly [the caching system may prescan the cache memory during prefetch to alter the prefetch amount is response to a command request (abstract); figures 8A-8E shows the details of adjusting the prefetch amount].
>
> Second, the citation "For every read command the invention determines how much data to prefetch after the requested data is retrieved (column 10, lines 62-67)" means determining the amount of data to be prefetched for the current read command after the requested data of the previous read command is retrieved.
>
> This is illustrated in more details in figures 10B-10F. In figure 10B, step 725 determines if this is a "read" command, if it is a "read" command, step B2 follows. In figure 10C, step B2, step 754 determines if this is a "partial hit in prefetch and prefetch from previous command will fetch a higher LBA than the current command." If the answer is "YES," the prefetch length is adjusted to accommodate data already requested, as stated in step 756.
>
> Third, the requested data and prefetch data are retrieved from the disk drive [figure 1, 40] at the same time rather than separately. This is illustrated in figure 1C. In figure 1C, Step 104 determines if there is a command from the host to be processed. Step 126 computes the prefetch length, followed by Step 132, which set buffer counter (to inform the disk and to monitor how much data to be transferred from the disk for this transfer operation) and start the disk (instruct the disk to begin transfer data). Note that in the entire flowchart only one step, Step 132, involves data transferring from the disk. Thus all data transfer from the disk to the cache buffer is performed only once, not twice for requested data and prefetch data separately. (Examiner's Answer at pages 15-17) (emphasis in original).

The Examiner's first two points above do not identify any disclosure in Hicken that teaches or suggests that the prefetch operation disclosed in Hicken is not a separate transfer. Rather, the Examiner again relies on Figure 1C as the only basis for the Examiner's contention that "the requested data and prefetch data are retrieved from the disk drive [figure 1, 40] at the same time rather than separately." Figure 1C of Hicken is a high level flow diagram that shows a method of processing commands from a host. As is typical with such high level diagrams, the various blocks in the diagram are represented in additional detail in later Figures, and involve multiple steps or operations. The conclusion that "all data transfer from the disk to the cache buffer is performed only once, not twice for requested data and prefetch data separately", which is based solely on the fact that only one step in this single

high level diagram mentions data transferring from the disk, is inconsistent with the more detailed disclosure provided throughout Hicken. For example, Hicken discloses with respect to Figure 16A that:

> The method starts by receiving a first command in step 1126. The invention is in hyper mode at this stage of processing. The invention then processes the first command and only caches requested data from the disk with no prefetch in step 1128. The process then receives a second command in step 1130, which is assumed to be a random command, and processes the second command in step 1132. In step 1134, a determination is made as to whether or not the second command is sequential with the first command. If the second command is sequential with the first command, the process flows to step 1136 to cache prefetch data from the second command and identifies the second command as sequential. (Hicken at col. 37, lines 33-44).

As the above disclosure and Figure 16A clearly indicate, the prefetch at 1136 is performed **after** the second command (i.e., current command) is processed at 1132. In addition, on page 10 of the Appeal Brief, Appellant cited numerous disclosures throughout the Hicken reference that all indicate that the prefetch is a separate operation. Nonetheless, as Appellant pointed out in the Appeal Brief, it is not Appellant's burden to prove whether the prefetch is a separate operation or not. It is the Examiner's burden to establish that the prior art teaches or suggests all of the limitations of the claims. None of the Examiner's above citations to Hicken teaches or suggests adding a prefetch value to a transfer length value specified in a current non-sequential read command, and then providing this sum to a data storage device, thereby requesting data specified by the current host command and prefetch data. In addition, the Examiner's new argument that these limitations are inherent in Hicken, implicitly acknowledges that Hicken does not include an explicit teaching or suggestion.

In view of the above, independent claim 1 is not taught or suggested by Hicken. Appellants submit that independent claim 1 is not anticipated by Hicken, and respectfully request that the rejection of independent claim 1 under 35 U.S.C. § 102(b) be withdrawn.

Dependent claims 2-8, which further limit patentably distinct claim 1, are also believed to be allowable over the cited reference. Appellants submit that dependent claims 2-8 are not anticipated by Hicken, and respectfully request that the rejection of dependent claims 2-8 under 35 U.S.C. § 102(b) be withdrawn.

The Examiner stated in the Response to Arguments section with respect to claims 9 and 10 that:

> Appellants also contend that, with respect to claim 9, the MIN and MAX prefetch values disclosed by Hicke [sic] are only "numbers" and not "registers." The Examiner disagrees.
>
> TABLE 1 (col. 7, lines 35-50) of Hicken shows that the MAX prefetch variable is represented using a register designated as CH-ENV-ADC-MAXPF, and that the MIN prefetch variable is represented using a register designated as CH-ENV-ADC-MINPF. Thus, MAX and MIN are registers storing the values of the maximum and minimum prefetch value, respectively, and their values may be dynamically adjusted depending on the access type [abstract].
>
> Appellants further contend that, with respect to claim 9, the Hicken reference fails to teach the element of multiplexer. The Examiner disagrees.
>
> First, Hicken teaches that the corresponding multiplexing function is shown in figure 8D: COMPUTE PREFETCH, where either the value of the MINPREFETCH register (step 692) or the value of the MAXPREFETCH register (step 694) is selected, depending on the condition of step 690;
>
> Second, Hicken shows that MINPREFETCH is set to BLOCKPERSEG in figure 6C, step 592 and that MINPREFETCH is set to 0 in figure 6C, steps 586 and 592. Thus the selection of either the value of the MINPREFETCH register (step 692) or the value of the MAXPREFETCH register (step 694) is to select a prefetch value that is either BLOCKPERSEG (when the MAXPREFETCH register is selected, step 694) or 0 (when the MINPREFETCH register is selected, step 692). Note also that a prefetch value of 0 simply means no prefetch.
>
> Third, Hicken shows that the default values are zero for the MIN and blocks per segment for the MAX; this allows the prefetch to be interrupted (i.e., no prefetch) as soon as a seek can be started for a new command, but fills the segment with new data if the prefetch is not interrupted (column 10, lines 62-67; column 11, lines 1-13).
>
> Therefore, Hicken clearly teaches "multiplexing" the prefetch values of BLOCKPERSEG and 0 by using the MAXPREFETCH register and the MINPREFETCH register. (Examiner's Answer at pages 17-19) (emphasis in original).

The MIN and MAX values identified by the Examiner are not registers that are coupled to a multiplexer. Figure 8D is a flow diagram, not a multiplexer coupled to registers. In addition, the Examiner indicated above that either the MIN value or the MAX value "is selected, depending on the condition of step 690". Step 690 in Figure 8D discloses "WAS HIT DETECTED IN PREFETCH OR NOT A FULL CACHE HIT". In contrast, dependent claim 9 recites that "the multiplexer responsive to the new sequential read indication for selectively outputting the prefetch value or the zero value." The "multiplexer" allegedly

disclosed in Hicken (i.e., the flow diagram of Figure 8D), is not responsive to a new sequential read indication.

Dependent claims 9 and 10, which further limit patentably distinct claim 1, are also believed to be allowable over the cited reference. Appellants submit that dependent claims 9 and 10 are not anticipated by Hicken, and respectfully request that the rejection of dependent claims 9 and 10 under 35 U.S.C. § 102(b) be withdrawn.

Dependent claim 11 recites "the prefetch controller of claim 10, wherein the transfer length generator further comprises: an adder for adding the value stored in the third register and the value output by the multiplexer." As Appellant pointed out in the Appeal Brief, the Examiner has not identified any structure in Hicken that corresponds to, or performs the functions of, the adder recited in claim 11. The Examiner has not identified any structure in Hicken that adds the transfer length value specified in a current host command to a prefetch value that has been selected and output by a multiplexer.

The Examiner stated in the Response to Arguments section with respect to claim 11 that:

> It is inherent that "an adding function" has to be performed in order to "adding a prefetch value to a transfer length value specified in a current non-sequential read command, and then providing this sum to a data storage device," as recited in claim 1. (Examiner's Answer at page 19).

As Appellant addressed above with respect to claim 1, inherent anticipation requires that the missing descriptive material is "necessarily present," not merely "probably or possibly present". The Examiner has not provided any rational underpinning to support the argument that Hicken must necessarily have an adder that adds the transfer length value specified in a current host command to a prefetch value that has been selected and output by a multiplexer. In fact, the numerous disclosures throughout Hicken that indicate that the prefetch is a separate transfer, teach away from using such an adder.

Dependent claim 11, which further limits patentably distinct claim 1, is also believed to be allowable over the cited reference. Appellants submit that dependent claim 11 is not anticipated by Hicken, and respectfully request that the rejection of dependent claim 11 under 35 U.S.C. § 102(b) be withdrawn.

The Examiner stated in the Response to Arguments section with respect to claims 12-16 that:

> Appellants contend that claims 12-16 are allowable over Hicken because, allegedly, Hicken fails to teach, or suggest "adding a prefetch value to a transfer length value specified in a current non-sequential read command, and then providing this sum to a data storage device." The Examiner disagrees. The Examiner has fully addressed and demonstrated that Hicken indeed teach the limitation of "adding a prefetch value to a transfer length value specified in a current non-sequential read command, and then providing this sum to a data storage device" in the previous section of this Office Action. Refer to ***"Response to Arguments (A) on Rejections of Claims 1-8"*** for details. (Examiner's Answer at page 20) (emphasis in original).

Independent claim 12 recites "adding a prefetch length value to the first transfer length value if the current read command and the previous read command are non-sequential, thereby generating a second transfer length value; and outputting the second transfer length value to the data storage device." For at least the reasons set forth above with respect to claim 1, Hicken does not teach or suggest the above-quoted limitations of claim 12.

In view of the above, independent claim 12 is not taught or suggested by Hicken. Appellants submit that independent claim 12 is not anticipated by Hicken, and respectfully request that the rejection of independent claim 12 under 35 U.S.C. § 102(b) be withdrawn.

Dependent claims 13-16, which further limit patentably distinct claim 12, are also believed to be allowable over the cited reference. Appellants submit that dependent claims 13-16 are not anticipated by Hicken, and respectfully request that the rejection of dependent claims 13-16 under 35 U.S.C. § 102(b) be withdrawn.

The Examiner stated in the Response to Arguments section with respect to claims 17-18 that:

> Appellants contend that claims 17-18 are allowable over Hicken because, allegedly, Hicken fails to teach, or suggest "adding a prefetch value to a transfer length value specified in a current non-sequential read command, and then providing this sum to a data storage device." The Examiner disagrees. The Examiner has fully addressed and demonstrated that Hicken indeed teach the limitation of "adding a prefetch value to a transfer length value specified in a current non-sequential read command, and then providing this sum to a data storage device" in the previous section of this Office Action. Refer to ***"Response to Arguments (A) on Rejections of Claims 1-8"*** for details. (Examiner's Answer at page 20) (emphasis in original).

Independent claim 17 recites "transfer length generation means for adding a prefetch length value to a transfer length value specified in the current host command if the current host command specifies a non-sequential read operation, the transfer length generation means configured to output a sum of the prefetch length value and the transfer length value to the storage means." For at least the reasons set forth above with respect to claim 1, Hicken does not teach or suggest the above-quoted limitations of claim 17.

In view of the above, independent claim 17 is not taught or suggested by Hicken. Appellants submit that independent claim 17 is not anticipated by Hicken, and respectfully request that the rejection of independent claim 17 under 35 U.S.C. § 102(b) be withdrawn.

Dependent claim 18, which further limits patentably distinct claim 17, is also believed to be allowable over the cited reference. Appellants submit that dependent claim 18 is not anticipated by Hicken, and respectfully request that the rejection of dependent claim 18 under 35 U.S.C. § 102(b) be withdrawn.

The Examiner stated in the Response to Arguments section with respect to claim 19 that:

> Appellants contend that claim 19 is allowable over Hicken because the Examiner has not identified any structure in Hicken that corresponds to, or performs the functions of the adding means recited in claim 19. The Examiner disagrees.
>
> Appellants contend that, with respect to claim 19, the MIN and MAX prefetch values disclosed by Hicke (sic) are only "numbers" and not "registers." The Examiner disagrees.
>
> TABLE 1 (col. 7, lines 35-50) of Hicken shows that the MAX prefetch variable is represented using a register designated as CH-ENV-ADC-MAXPF, and that the MIN prefetch variable is represented using a register designated as CH-ENV-ADC-MINPF. Thus, MAX and MIN are registers storing the values of the maximum and minimum prefetch value, respectively, and their values may be dynamically adjusted depending on the access type [abstract].
>
> Appellants further contend that, with respect to claim 19, the Hicken reference fails to teach the element of multiplexer. The Examiner disagrees.
>
> First, Hicken teaches that the corresponding multiplexing function is shown in figure 8D: COMPUTE PREFETCH, where either the value of the MINPREFETCH register (step 692) or the value of the MAXPREFETCH register (step 694) is selected, depending on the condition of step 690;
>
> Second, Hicken shows that MINPREFETCH is set to BLOCKPERSEG in figure 6C, step 592 and that MINPREFETCH is set to 0 in figure 6C, steps 586 and 592. Thus the selection of either the value of the

> MINPREFETCH register (step 692) or the value of the MAXPREFETCH register (step 694) is to select a prefetch value that is either BLOCKPERSEG (when the MAXPREFETCH register is selected, step 694) or 0 (when the MINPREFETCH register is selected, step 692). Note also that a prefetch value of 0 simply means no prefetch.
>
> Third, Hicken shows that the default values are zero for the MIN and blocks per segment for the MAX; this allows the prefetch to be interrupted (i.e., no prefetch) as soon as a seek can be started for a new command, but fills the segment with new data if the prefetch is not interrupted (column 10, lines 62-67; column 11, lines 1-13).
>
> Therefore, Hicken clearly teaches "multiplexing" the prefetch values of BLOCKPERSEG and 0 by using the MAXPREFETCH register and the MINPREFETCH register.
>
> It is inherent that "an adding function" has to be performed in order to "adding a prefetch value to a transfer length value specified in a current non-sequential read command, and then providing this sum to a data storage device," as recited in claim 17.
>
> Therefore, the Examiner's position regarding the patentability of claim 19 remains the same as indicated in the previous Office Action. (Examiner's Answer at pages 21-22) (emphasis in original).

Dependent claim 19 recites "the memory device of claim 17, wherein the transfer length generation means comprises: first register means for storing the prefetch length value; second register means for storing a zero value; multiplexing means for selectively outputting the prefetch length value or the zero value based on an output of the sequential read detection means; and adding means for adding an output of the multiplexing means and the transfer length value specified in the current host command."

Figure 8D is a flow diagram, not a multiplexing means. In addition, the Examiner indicated above that either the MIN value or the MAX value "is selected, depending on the condition of step 690". Step 690 in Figure 8D discloses "WAS HIT DETECTED IN PREFETCH OR NOT A FULL CACHE HIT". In contrast, dependent claim 19 recites "multiplexing means for selectively outputting the prefetch length value or the zero value based on an output of the sequential read detection means." The "multiplexing means" allegedly disclosed in Hicken (i.e., the flow diagram of Figure 8D), does not selectively output a prefetch length value or a zero value based on an output of a sequential read detection means.

The Examiner also has not identified any structure in Hicken that corresponds to, or performs the functions of, the adding means recited in claim 19. The Examiner has not identified any structure in Hicken that adds the transfer length value specified in a current host command to a prefetch value or zero value that has been output by a multiplexing means. These limitations are also not inherent in Hicken. As Appellant addressed above with respect to claim 1, inherent anticipation requires that the missing descriptive material is "necessarily present," not merely "probably or possibly present". The Examiner has not provided any rational underpinning to support the argument that Hicken must necessarily have an adding means that adds an output of a multiplexing means and the transfer length specified in a current host command. In fact, the numerous disclosures throughout Hicken that indicate that the prefetch is a separate transfer, teach away from using such an adding means.

Dependent claim 19, which further limits patentably distinct claim 17, is also believed to be allowable over the cited reference. Appellants submit that dependent claim 19 is not anticipated by Hicken, and respectfully request that the rejection of dependent claim 19 under 35 U.S.C. § 102(b) be withdrawn.

The Examiner stated in the Response to Arguments section with respect to claims 20-27 that:

> Appellants contend that claims 20-27 are allowable over Hicken because, allegedly, Hicken fails to teach, or suggest "adding a prefetch value to a transfer length value specified in a current non-sequential read command, and then providing this sum to a data storage device." The Examiner disagrees.
>
> The Examiner has fully addressed and demonstrated that Hicken indeed teach the limitation of "adding a prefetch value to a transfer length value specified in a current non-sequential read command, and then providing this sum to a data storage device" in the previous section of this Office Action. Refer to ***"Response to Arguments (A) on Rejections of Claims 1-8"*** for details. (Examiner's Answer at pages 22-23) (emphasis in original).

Independent claim 20 recites "generating a new sequential read indication for the current host command if the current host command and a previously received host command specify read operations that are non-sequential; outputting a first transfer length value to the data storage device if the new sequential read indication is generated for the current host

command, wherein the first transfer length value is determined by adding a prefetch value to a transfer length value specified in the current host command; and outputting a second transfer length value to the data storage device if the new sequential read indication is not generated for the current host command, the second transfer length value less than the first transfer length value." For at least the reasons set forth above with respect to claim 1, Hicken does not teach or suggest the above-quoted limitations of claim 20.

In view of the above, independent claim 20 is not taught or suggested by Hicken. Appellants submit that independent claim 20 is not anticipated by Hicken, and respectfully request that the rejection of independent claim 20 under 35 U.S.C. § 102(b) be withdrawn.

Dependent claims 21-27, which further limit patentably distinct claim 20, are also believed to be allowable over the cited reference. Appellants submit that dependent claims 21-27 are not anticipated by Hicken, and respectfully request that the rejection of dependent claims 21-27 under 35 U.S.C. § 102(b) be withdrawn.

The Examiner stated in the Response to Arguments section with respect to claims 28-29 that:

> Appellants contend that claim 11 is allowable over Hicken because there is no teaching or suggestion in Hicken regarding selectively outputting a prefetch value or a zero value based on whether a new sequential read indication is generated for the current host command. The Examiner disagrees due to the following reasons:
>
> First, Hicken teaches that "each segment is monitored to determine **access types** such as sequential, random, and repeating. The **access type** determines the amount of data to prefetch and to save, including a minimum and maximum prefetch (abstract)."
>
> Second, Hicken further teaches that "If this cache **access** is a sequential stream, both min and max prefetch are be set to the number of blocks in a segment to fill a segment's worth of data. The system then discards requested data for this command once the data has been transferred. If this cache **access** is a repetitive type of access, the min is set to the blockcount of the command and the max is the number of blocks in a segment less the blockcount of the command in order to keep the requested data for possible repeated access in subsequent commands. **The default values are zero** for the min and blocks per segment for the max; this allows the prefetch to be interrupted as soon as a seek can be started for a new command, but fills the segment with new data if the prefetch is not interrupted (column 10, lines 62-67; column 11, lines 1-13)." (Examiner's Answer at pages 23-24) (emphasis in original).

Dependent claim 28 recites "the computer-readable medium of claim 20, wherein the method further comprises: storing a prefetch value; storing a zero value; and selectively outputting the prefetch value or the zero value based on whether the new sequential read indication is generated for the current host command." Independent claim 20 indicates that the new sequential read indication is generated if the current host command and a previously received host command specify read operations that are non-sequential. The above cited-portions of Hicken relate to repetitive, sequential accesses. There is no teaching or suggestion in Hicken regarding selectively outputting a prefetch value or a zero value based on whether a new sequential read indication is generated for the current host command.

Dependent claims 28 and 29, which further limit patentably distinct claim 20, are also believed to be allowable over the cited reference. Appellants submit that dependent claims 28 and 29 are not anticipated by Hicken, and respectfully request that the rejection of dependent claims 28 and 29 under 35 U.S.C. § 102(b) be withdrawn.

The Examiner stated in the Response to Arguments section with respect to claim 30 that:

> Appellants contend that claim 30 is allowable over Hicken because, allegedly, Hicken fails to teach, or suggest "adding a prefetch value to a transfer length value specified in a current non-sequential read command, and then providing this sum to a data storage device." The Examiner disagrees.
>
> The Examiner has fully addressed and demonstrated that Hicken indeed teach the limitation of "adding a prefetch value to a transfer length value specified in a current non-sequential read command, and then providing this sum to a data storage device" in the previous section of this Office Action. Refer to **"Response to Arguments (A) on Rejections of Claims *1-8*"** for details. (Examiner's Answer at page 24) (emphasis in original).

Dependent claim 30 recites "the computer-readable medium of claim 29, wherein the method further comprises: adding the stored transfer length value and the selectively output value." The Examiner has not identified any structure in Hicken that adds the transfer length value specified in a current host command to a selectively output prefetch value or zero value.

Dependent claim 30, which further limits patentably distinct claim 20, is also believed to be allowable over the cited reference. Appellants submit that dependent claim 30 is not

anticipated by Hicken, and respectfully request that the rejection of dependent claim 30 under 35 U.S.C. § 102(b) be withdrawn.

CONCLUSION

For the above reasons, Appellant respectfully submits that the art of record neither anticipates nor renders obvious the claimed invention. Thus, the claimed invention does patentably distinguish over the art of record. Appellant, therefore, respectfully submits that the above rejections are not correct and should be withdrawn, and respectfully requests that the Examiner be reversed and that all pending claims be allowed.

Any inquiry regarding this Response should be directed to Jeff A. Holmen at Telephone No. (612) 573-0178, Facsimile No. (612) 573-2005. In addition, all correspondence should continue to be directed to the following address:

IP Administration
Legal Department, M/S 35
HEWLETT-PACKARD COMPANY
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Respectfully submitted,

Robin Alexis Takasugi et al.

By their attorneys,

DICKE, BILLIG & CZAJA, PLLC
Fifth Street Towers, Suite 2250
100 South Fifth Street
Minneapolis, MN 55402
Telephone: (612) 573-0178
Facsimile: (612) 573-2005

Date: **June 4, 2008**
JAH:jms

/Jeff A. Holmen/
Jeff A. Holmen
Reg. No. 38,492